Toward Faster Implicit Network Training via Random Truncation

SIAM OP26

Brandon Knutson, Krishna Balasubramanian, Samy Wu Fung, Daniel McKenzie

COLORADO SCHOOL OF MINES

What is an implicit network?

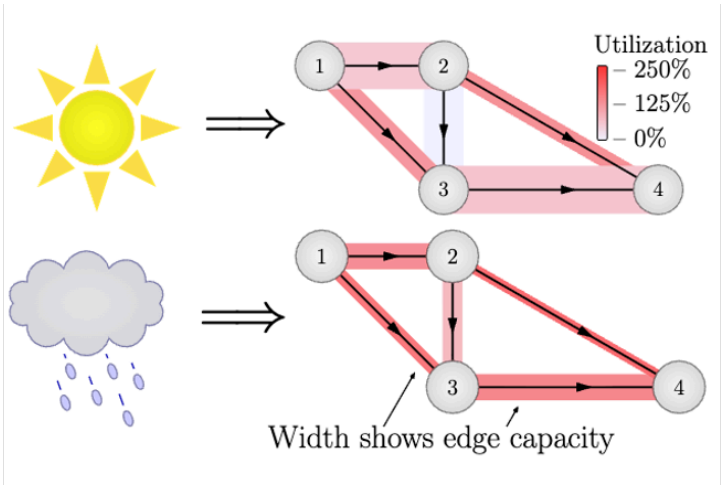
An **implicit neural network (INN)** is defined by a parametrized function F whose fixed point x^* is the network's output.

$$x^*(\theta, d) = F(x^*(\theta, d), \theta, d)$$

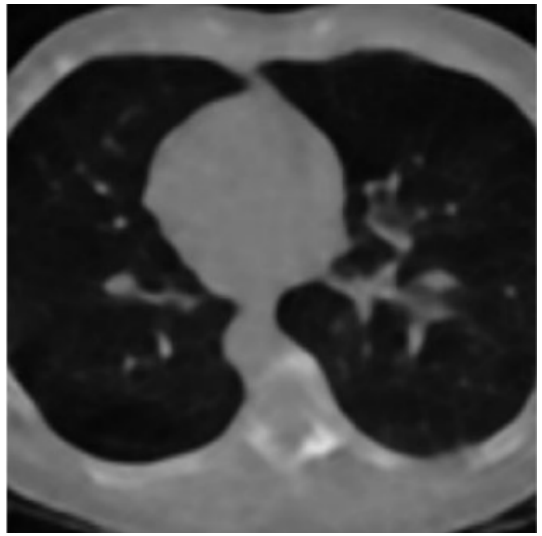
- $x^*(\theta, d)$: fixed point
- θ: parameters
- d: data

Applications of INNs

Mazes
(Knutson et al., 2026)



Traffic flow
(McKenzie et al., 2024)



Medical imaging
(Heaton et al., 2021)

Training an INN on data $d \sim P$ is a **bilevel** optimization problem:

$$\begin{aligned} &\min_{\theta} L(\theta) := \mathbb{E}_d \, \ell(x^\star(\theta, d), d) \\ \text{s.t.} \quad &x^\star(\theta, d) = F(x^\star(\theta, d), \theta, d) \quad \forall \theta, d \end{aligned} \tag{INN-TP}$$

Stochastic gradient method (SGM)

Input: initial θ^1, step size $\alpha > 0$, iterations T
for $t = 1, \ldots, T$ **do**
    Sample $d^t \sim \mathbb{P}_d$
    Compute stochastic gradient estimate $v(\theta^t, d^t)$
    $\theta^{t+1} \leftarrow \theta^t - \alpha \cdot v(\theta^t, d^t)$
end for
Sample $\hat{\theta}^T$ uniformly from $\{\theta^1, \ldots, \theta^T\}$
Output: $\hat{\theta}^T$

Hypergradient

- Solving (INN-TP) with SGM requires estimating the **hypergradient** $\nabla L(\theta)$:

$$
\begin{aligned}
\nabla L(\theta) &= \nabla_\theta \, \mathbb{E}_d \, \ell(x^\star(\theta, d), d) && \text{(definition of } L\text{)} \\
&= \mathbb{E}_d \, \nabla_\theta \ell(x^\star(\theta, d), d) && \text{(Leibniz)} \\
&= \mathbb{E}_d \underbrace{\nabla_1 \ell(x^\star, d) \cdot (I - J_1 F(x^\star, \theta, d))^{-1} \cdot J_2 F(x^\star, \theta, d)}_{=: \nabla_\theta L(\theta, d)} && \text{(implicit differentiation)}
\end{aligned}
$$

Hypergradient

- Solving (INN-TP) with SGM requires estimating the **hypergradient** $\nabla L(\theta)$:

$$
\begin{aligned}
\nabla L(\theta) &= \nabla_\theta \, \mathbb{E}_d \, \ell(x^\star(\theta, d), d) && \text{(definition of } L\text{)} \\
&= \mathbb{E}_d \, \nabla_\theta \ell(x^\star(\theta, d), d) && \text{(Leibniz)} \\
&= \mathbb{E}_d \underbrace{\nabla_1 \ell(x^\star, d) \cdot (I - J_1 F(x^\star, \theta, d))^{-1} \cdot J_2 F(x^\star, \theta, d)}_{=: \nabla_\theta L(\theta, d)} && \text{(implicit differentiation)}
\end{aligned}
$$

- $\nabla_\theta L(\theta, d)$ is the **sample-wise hypergradient**

Existing INN hypergradient estimators

Approximate the **sample-wise hypergradient**:

(1) K: fixed point solver iterations ($x^\star \approx x^K$)

(2) N: inverse approximation order

Existing INN hypergradient estimators

Approximate the **sample-wise hypergradient**:

(1) K: fixed point solver iterations ($x^\star \approx x^K$)

(2) N: inverse approximation order

True	$\nabla_\theta L(\theta, d) :=$	$\nabla_1 \ell(x^\star, d)$	$(I - J_1 F(x^\star))^{-1}$	$J_2 F(x^\star)$
1) AD	$v_{\text{AD}}^K :=$	$\nabla_1 \ell(\boldsymbol{x^K})$	$J_\theta \boldsymbol{x^K}$ (backprop)	
2) ID	$v_{\text{ID}}^{K,N} :=$	$w^N(\boldsymbol{x^K})$ (linear solve)		$J_2 F(\boldsymbol{x^K})$
3) Neu	$v_{\text{Neu}}^{K,N} :=$	$\nabla_1 \ell(\boldsymbol{x^K})$	$\sum_{n=0}^N (J_1 F(\boldsymbol{x^K}))^n$	$J_2 F(\boldsymbol{x^K})$
4) JFB	$v_{\text{JFB}}^K :=$	$\nabla_1 \ell(\boldsymbol{x^K})$	I	$J_2 F(\boldsymbol{x^K})$

Motivating question

*Can we speed INN training by **randomly truncating** the number of iterations {1, ..., K} used to approximate the fixed point?*

*Can we speed INN training by **randomly truncating** the number of iterations {1, ..., K} used to approximate the fixed point?*

- Random truncation has **long history**: computational physics (Forsythe & Leibler, 1950), computer graphics (Arvo & Kirk, 1990), multilevel Monte Carlo (Giles, 2015), etc.

Motivating question

*Can we speed INN training by **randomly truncating** the number of iterations {1, ..., K} used to approximate the fixed point?*

- Random truncation has **long history**: computational physics (Forsythe & Leibler, 1950), computer graphics (Arvo & Kirk, 1990), multilevel Monte Carlo (Giles, 2015), etc.
- Recent advances in **bilevel optimization**: (Beatson & Adams, 2019; Hu et al., 2024)

We prove:

(1) SGM with INN hypergradient estimators achieve $\tilde{O}(\varepsilon^{-4})$ complexity .[1]

(2) RT reduces AD and JFB per-step cost from $O(\log \varepsilon^{-1})$ to $O(\log \log \varepsilon^{-1})$.

[1] Matches the single-level non-convex stochastic optimization lower bound (Arjevani et al., 2023).

Random truncation (RT)

(1) **Assume** estimators $v^0, v^1, \ldots, v^J$ at increasing levels.

Random truncation (RT)

(1) **Assume** estimators $v^0, v^1, ..., v^J$ at increasing levels.

(2) **Choose** a distribution P on $\{1, ..., J\}$ with $p_j > 0$ for all $j \in \{1, ..., J\}$.

Random truncation (RT)

(1) **Assume** estimators $v^0, v^1, ..., v^J$ at increasing levels.

(2) **Choose** a distribution P on $\{1, ..., J\}$ with $p_j > 0$ for all $j \in \{1, ..., J\}$.

(3) **Telescope and rewrite** as expectation over $j \sim P$:

$$v^J = v^0 + \sum_{j=1}^{J} (v^j - v^{j-1})$$

Random truncation (RT)

(1) **Assume** estimators $v^0, v^1, \ldots, v^J$ at increasing levels.

(2) **Choose** a distribution P on $\{1, \ldots, J\}$ with $p_j > 0$ for all $j \in \{1, \ldots, J\}$.

(3) **Telescope and rewrite** as expectation over $j \sim P$:

$$v^J = v^0 + \sum_{j=1}^{J} (v^j - v^{j-1}) = v^0 + \sum_{j=1}^{J} p_j \cdot \frac{v^j - v^{j-1}}{p_j}$$

Random truncation (RT)

(1) **Assume** estimators $v^0, v^1, \ldots, v^J$ at increasing levels.

(2) **Choose** a distribution P on $\{1, \ldots, J\}$ with $p_j > 0$ for all $j \in \{1, \ldots, J\}$.

(3) **Telescope and rewrite** as expectation over $j \sim P$:

$$v^J = v^0 + \sum_{j=1}^{J} (v^j - v^{j-1}) = v^0 + \sum_{j=1}^{J} p_j \cdot \frac{v^j - v^{j-1}}{p_j} = v^0 + \mathbb{E}_{j \sim P} \frac{v^j - v^{j-1}}{p_j}$$

Random truncation (RT)

(1) **Assume** estimators $v^0, v^1, \ldots, v^J$ at increasing levels.

(2) **Choose** a distribution P on $\{1, \ldots, J\}$ with $p_j > 0$ for all $j \in \{1, \ldots, J\}$.

(3) **Telescope and rewrite** as expectation over $j \sim P$:

$$v^J = v^0 + \sum_{j=1}^{J} (v^j - v^{j-1}) = v^0 + \sum_{j=1}^{J} p_j \cdot \frac{v^j - v^{j-1}}{p_j} = v^0 + \mathbb{E}_{j \sim P} \frac{v^j - v^{j-1}}{p_j}$$

(4) **Define** the **RT estimator** by replacing $\mathbb{E}_{j \sim P}$ with a single sample $j \sim P$:

$$\hat{v}^J := v^0 + \frac{v^j - v^{j-1}}{p_j}.$$

Doubly exponential sampling

Following (Hu et al., 2024):

(1) We set $p_j \propto 2^{-j}$ for $j \in \{1, \ldots, J\}$

(2) We iterate $k = 2^j$ times at level j (max iterations $K = 2^J$)

Doubly exponential sampling

Following (Hu et al., 2024):

(1) We set $p_j \propto 2^{-j}$ for $j \in \{1, \ldots, J\}$

(2) We iterate $k = 2^j$ times at level j (max iterations $K = 2^J$)

$\Rightarrow$ **Unbiased**: $\mathbb{E}_j[\hat{v}^J] = v^J$

Doubly exponential sampling

Following (Hu et al., 2024):

(1) We set $\boldsymbol{p_j \propto 2^{-j}}$ for $j \in \{1, \ldots, J\}$

(2) We iterate $\boldsymbol{k = 2^j}$ times at level j (max iterations $K = 2^J$)

$\Rightarrow$ **Unbiased**: $\mathbb{E}_j[\hat{v}^J] = v^J$

$\Rightarrow$ **Fewer iterations**: $\mathbb{E}_j[2^j] = \sum_{j=1}^{J} p_j \cdot 2^j \approx J = \log_2(K) < K$

Base and RT estimators

Base estimators: v_{AD}^{K} $v_{\mathrm{ID}}^{K,N}$ $v_{\mathrm{Neu}}^{K,N}$ v_{JFB}^{K}

RT estimators: $\hat{v}_{\mathrm{AD}}^{J}$ $\hat{v}_{\mathrm{ID}}^{J,N}$ $\hat{v}_{\mathrm{Neu}}^{J,N}$ $\hat{v}_{\mathrm{JFB}}^{J}$

Assumptions

1) Model regularity

- F is ρ-contractive in x (with $\rho < 1$)
- F, $J_1 F$, $J_2 F$, ℓ, $\nabla_1 \ell$ all Lipschitz
- Bounded initial iterate: $\left\| x^0 - x^\star \right\| \leq R$

[1]Follows from cheap gradient principle (Griewank & Walther, 2008).

Assumptions

1) Model regularity

- F is ρ-contractive in x (with $\rho < 1$)
- F, $J_1 F$, $J_2 F$, ℓ, $\nabla_1 \ell$ all Lipschitz
- Bounded initial iterate: $\left\| x^0 - x^\star \right\| \leq R$

2) Stochastic regularity

- Fixed-point variance: $\mathrm{Var}_{d_1}\left(x^\star(\theta, d_1)\right) \leq \sigma_1^2$
- Sample-gradient variance: $\mathrm{Var}_{d_2 \mid d_1}\left(\nabla_1 \ell\right) \leq \sigma_2^2$
- Lipschitz conditional mean and Jacobian of $x^\star$

[1]Follows from cheap gradient principle (Griewank & Walther, 2008).

Assumptions

1) Model regularity

- F is ρ-contractive in x (with $\rho < 1$)
- F, $J_1 F$, $J_2 F$, ℓ, $\nabla_1 \ell$ all Lipschitz
- Bounded initial iterate: $\left\| x^0 - x^\star \right\| \leq R$

2) Stochastic regularity

- Fixed-point variance: $\mathrm{Var}_{d_1}\left(x^\star(\theta, d_1)\right) \leq \sigma_1^2$
- Sample-gradient variance: $\mathrm{Var}_{d_2 \mid d_1}\left(\nabla_1 \ell\right) \leq \sigma_2^2$
- Lipschitz conditional mean and Jacobian of $x^\star$

3) Computational costs

- F evaluation: C_F operations
- Gradient evaluation: ωC_F operations[1]

[1] Follows from cheap gradient principle (Griewank & Walther, 2008).

Pointwise estimator errors

For every θ and d, each base estimator $\bullet \in \{\mathrm{AD}, \mathrm{ID}, \mathrm{Neu}, \mathrm{JFB}\}$ satisfies

$$\|v_\bullet(\theta, d) - \nabla L(\theta, d)\| \le b_\bullet$$

where :

$$b^k_{\mathrm{AD}} := \left(C_L + \tfrac{L_\nabla L_F R}{1-\rho}\right)\rho^k + \left(C_L L_{J_1} + L_\ell L_{J_2}\right) R\, k\rho^{k-1}$$

$$b^{k,N}_{\mathrm{ID}} := C_L \rho^N + \tfrac{L_\nabla L_F + L_\ell L_{J_2} + C_L L_{J_1}}{1-\rho} R\rho^k$$

$$b^k_{\mathrm{JFB}} := C_L \rho + \left(L_\nabla L_F + L_\ell L_{J_2}\right) R\rho^k$$

$$b^{k,N}_{\mathrm{Neu}} := C_L \rho^{N+1} + \left(\tfrac{1-\rho^{N+1}}{1-\rho}\left(L_\nabla L_F + L_\ell L_{J_2}\right) + C_L L_{J_1} \tfrac{1-(N+1)\rho^N + N\rho^{N+1}}{1-\rho}\right) R\rho^k$$

Pointwise estimator errors

For every θ and d, each base estimator $\bullet \in \{\mathrm{AD}, \mathrm{ID}, \mathrm{Neu}, \mathrm{JFB}\}$ satisfies

$$\|v_\bullet(\theta, d) - \nabla L(\theta, d)\| \le b_\bullet$$

where **as $k, N \to \infty$**:

$$b_{\mathrm{AD}}^k := \left(C_L + \tfrac{L_\nabla L_F R}{1-\rho}\right)\rho^k + \left(C_L L_{J_1} + L_\ell L_{J_2}\right) R\, k\rho^{k-1} \qquad \to 0$$

$$b_{\mathrm{ID}}^{k,N} := C_L \rho^N + \tfrac{L_\nabla L_F + L_\ell L_{J_2} + C_L L_{J_1}}{1-\rho} R\rho^k \qquad \to 0$$

$$b_{\mathrm{JFB}}^k := C_L\rho + \left(L_\nabla L_F + L_\ell L_{J_2}\right) R\rho^k \qquad \to C_L\rho$$

$$b_{\mathrm{Neu}}^{k,N} := C_L \rho^{N+1} + \left(\tfrac{1-\rho^{N+1}}{1-\rho}\left(L_\nabla L_F + L_\ell L_{J_2}\right) + C_L L_{J_1} \tfrac{1-(N+1)\rho^N + N\rho^{N+1}}{1-\rho}\right) R\rho^k \qquad \to 0$$

For every estimator $\bullet$,

$$\text{Bias} \leq b_\bullet$$

$$\text{MSE} \leq 2\, b_\bullet^2 + \underbrace{2\,\sigma_\nabla^2}_{\substack{\text{hypergradient} \\ \text{variance}}} + \underbrace{\sigma_\bullet^2}_{\substack{\text{RT level-sampling} \\ \text{variance}}}$$

Theorem (SGM convergence). Under the regularity assumptions, for any $\varepsilon > 0$, with hyperparameters

$$\begin{aligned} \alpha &= O(\varepsilon^2) && \text{step size} \\ N &= O(\log \varepsilon^{-1}) && \text{inverse approximation order (ID, Neu)} \\ K &= O(\log \varepsilon^{-1}) && \text{max lower-level iterations} \\ T &= O(\varepsilon^{-4}) && \text{upper-level iterations} \end{aligned}$$

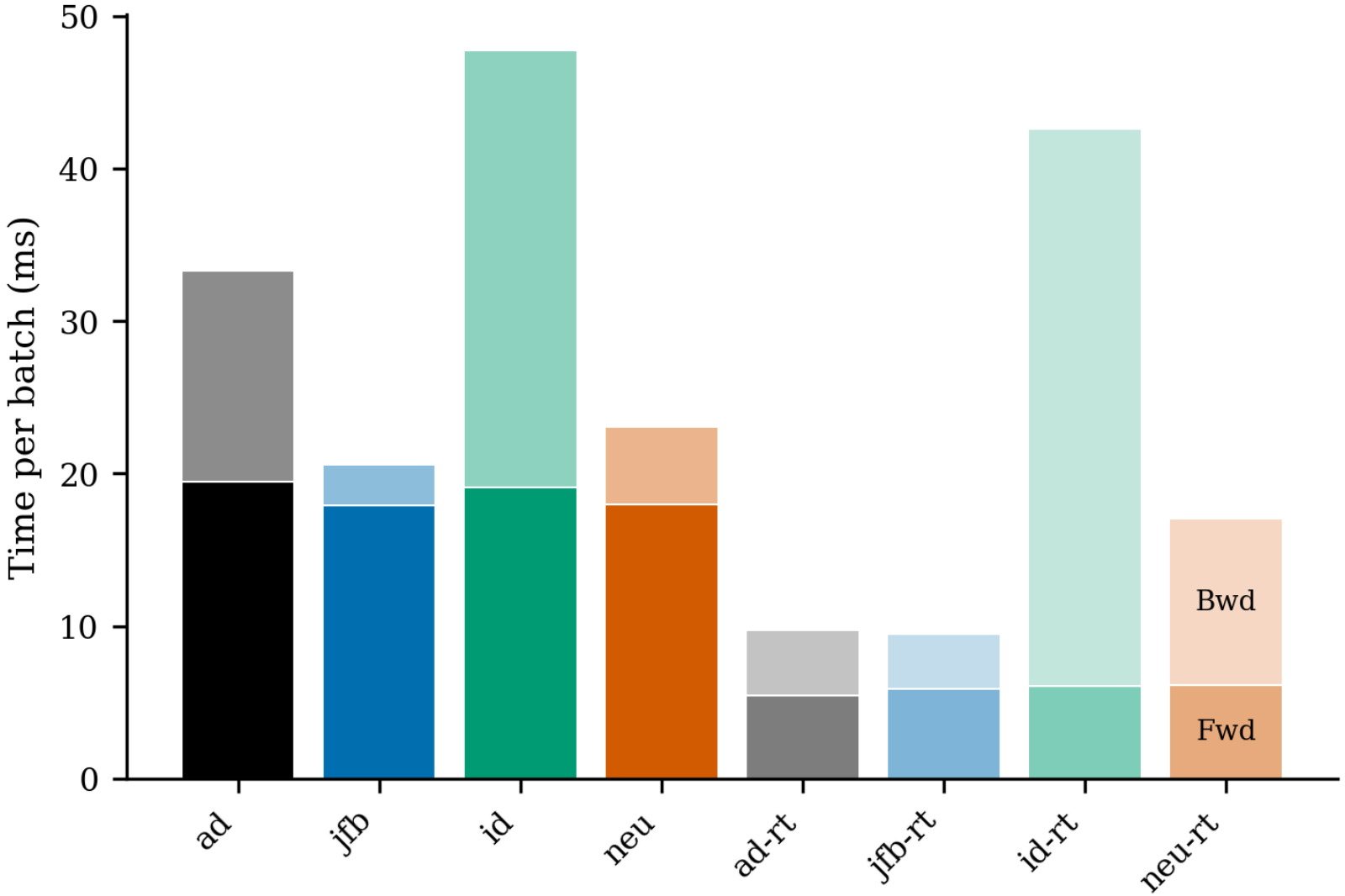
SGM satisfies … (next slide)

Complexity table

Estimator	Memory	Operations	Stationarity
AD	$O(\log(\varepsilon^{-1}))$	$O(\log(\varepsilon^{-1})\varepsilon^{-4})$	ε^2
ID	$O(\log(\varepsilon^{-1}))$	$O(\log^2(\varepsilon^{-1})\varepsilon^{-4})$	ε^2
JFB	$O(1)$	$O(\log(\varepsilon^{-1})\varepsilon^{-4})$	$2C_L^2\rho + \varepsilon^2$
Neu	$O(1)$	$O(\log(\varepsilon^{-1})\varepsilon^{-4})$	ε^2
AD-RT	$O(\log(\varepsilon^{-1}))$	$O(\log\log(\varepsilon^{-1})\,\varepsilon^{-4})$	ε^2
ID-RT	$O(\log(\varepsilon^{-1}))$	$O(\log^2(\varepsilon^{-1})\varepsilon^{-4})$	ε^2
JFB-RT	$O(1)$	$O(\log\log(\varepsilon^{-1})\,\varepsilon^{-4})$	$2C_L^2\rho + \varepsilon^2$
Neu-RT	$O(1)$	$O(\log(\varepsilon^{-1})\varepsilon^{-4})$	ε^2

Memory excludes the shared baseline. Operations are upper bounds on the expectation. Stationarity is the upper bound on $\mathbb{E}\,\|\nabla L\left(\hat{\theta}^T\right)\|^2$.

Preliminary experimental results



We prove:

(1) SGM with INN hypergradient estimators achieve $\tilde{O}(\varepsilon^{-4})$ complexity.

(2) RT reduces AD and JFB per-step cost from $O(\log \varepsilon^{-1})$ to $O(\log \log \varepsilon^{-1})$.

References

Arjevani, Y., Carmon, Y., Duchi, J. C., Foster, D. J., Srebro, N., & Woodworth, B. (2023). Lower Bounds for Non-Convex Stochastic Optimization. *Mathematical Programming*, *199*(1–2), 165–214. https://doi.org/10.1007/s10107-022-01822-7

Arvo, J., & Kirk, D. (1990). Particle transport and image synthesis. *Proceedings of the 17th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '90)*, 63–66.

Beatson, A., & Adams, R. P. (2019). Efficient Optimization of Loops and Limits with Randomized Telescoping Sums. *International Conference on Machine Learning*, *97*, 534–543.

Forsythe, G. E., & Leibler, R. A. (1950). Matrix inversion by a Monte Carlo method. *Mathematical Tables and Other Aids to Computation*, *4*(31), 127–129.

Giles, M. B. (2015). Multilevel Monte Carlo methods. *Acta Numerica*, *24*, 259–328.

Griewank, A., & Walther, A. (2008). *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation* (2nd ed.). SIAM.

Heaton, H., Wu Fung, S., Gibali, A., & Yin, W. (2021). Feasibility-based fixed point networks. *Fixed Point Theory and Algorithms for Sciences and Engineering*, *2021*(1).

References

Hu, Y., Wang, J., Xie, Y., Krause, A., & Kuhn, D. (2024). Contextual Stochastic Bilevel Optimization. *Advances in Neural Information Processing Systems*, *36*.

Knutson, B., Rabeendran, A. C., Ivanitskiy, M., Pettyjohn, J., Behn, C. D., Fung, S. W., & McKenzie, D. (2026). On logical extrapolation for mazes with recurrent and implicit networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, *40*, 22635–22643.

McKenzie, D., Heaton, H., Li, Q., Wu Fung, S., Osher, S., & Yin, W. (2024). Three-operator splitting for learning to predict equilibria in convex games. *SIAM Journal on Mathematics of Data Science*, *6*(3), 627–648.

Appendix: comparison with Arjevani lower bound

Three differences between our analysis and (Arjevani et al., 2023):

(1) **Cost unit.** We count FLOPs; Arjevani counts gradient-oracle calls. The coefficient of ωC_F in our FLOP expressions is the gradient-evaluation complexity.

(2) **Upper/lower gradient.** Arjevani counts upper-level $\nabla_\theta L$ queries; our ωC_F is for one lower-level VJP of F.

(3) **Stationarity.** JFB and JFB-RT reach only $\left(2C_L^2\rho + \varepsilon^2\right)$-stationarity.

Claim. $\nabla L(\theta) = \mathbb{E}_d[\nabla_\theta L(\theta, d)]$.

Proof. By the chain rule and fixed-point sensitivity (Model regularity),

$$\|\nabla_\theta L(\theta, d)\| = \|\nabla_1 \ell(x^\star, d_2) \cdot J_\theta x^\star(\theta, d_1)\| \leq L_\ell \cdot \frac{L_F}{1-\rho} =: C_L$$

uniformly in d. Since the constant C_L is integrable under $\mathbb{P}_d$, dominated convergence gives

$$\nabla_\theta \mathbb{E}_d[\ell(x^\star, d)] = \mathbb{E}_d[\nabla_\theta \ell(x^\star, d)]. \quad \square$$

Appendix: generality of fixed-point problems

$$\underbrace{L(x) = R(x)}_{\text{Solve equation}} \quad \Leftrightarrow \quad \underbrace{(L - R)(x) = 0}_{\text{Find root}} \quad \Leftrightarrow \quad \underbrace{(L - R + I)(x) = x}_{\text{Find fixed point}}$$